REMARKS

Claims 1, 4-10, 12, 14 and 16-18 remain pending in the present application. In view of the above amendments and the following remarks, it is respectfully submitted that all of the presently pending claims are allowable.

The Examiner has objected to claim 17 because of a typographical error. (*Office Action*, paragraph 5.) Claim 17 has been amended to correct the mistake. Thus, applicants respectfully request the Examiner to withdraw the objection to claim 17.

The Examiner has rejected each of the pending claims under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,006,018 to Burnett et al. ("Burnett"). (*Office Action*, paragraph 7.) Burnett discloses a file system translator to enable different distributed file systems to coexist and interoperate within a distributed computing environment domain. (*Burnett*, col. 2, lines 21 - 23). In particular, Burnett teaches a method for allowing a Network File System (NFS) Distributed Computing Environment (DCE) to operate with a Distributed File Service (DFS) DCE. Burnett allows NFS users (e.g., an NFS server) authenticated access to DFS (e.g., DFS client) so that they may access data with access controls. (*Id.*, col. 6, lines 36 - 38). The access to the DFS client by the NFS server is provided by using a translator 20 comprising a kernel portion 22 having an intermediate authenticator 24, an NFS listener 26, and a DFS remote attribute interface (DFSRAI) 28. (*Id.*, col. 6, lines 42 - 44). The authenticator 24 maintains "mappings between NFS and DFS, transforms incoming NFS credentials into suitable DFS credentials, contains the intermediate authenticator's remote attribute manager for [file path variables, such as] @sys and @host support." (*Id.*, col. 6, lines 60 - 65). The @sys variable indicates a hardware and/or software combination for a platform and @host is used in a diskless component. (*Id.*, col. 5 line 64 - col. 6, lines 7). The DFSRAI 28 registers its remote attribute manager in order to provide operations for extracting @sys and @host information. (*Id.*, col. 7, lines 10 - 12).

Once the NFS server requests access from the DFS client, the client attempts to obtain a DFSRAI handle for the request. After the handle is returned, then DFSRAI 28 is used to obtain @sys and @host values for the request. (*Id.*, col. 7, lines 16 - 19). DFSRAI 28 provides a mechanism in which a resource manager and objects producer (e.g., authenticator 24) registers itself and its methods which can be used to extract attributes (i.e., @sys and @host variables) associated with the object (i.e.,credential). Specifically, object consumers (i.e., DFS clients) can use DFSRAI 28 to locate a reference to the resource manager's methods and extract attributes after the proper handle has been received. (*Id.*, col. 7, lines 52 - 58). The object consumer can, as a result, call the resource manager's methods directly to extract the attributes associated with the object. (*Id.*, col. 7, lines 66 - 67).

Independent claim 1 of the present invention recites "creating a producer component including a data object and a component module, the component module including information identifying the data object and an object handler to interact with the data object." As described in the specification, each producer component includes a data object and a component module (*e.g.*, an IDB+ module). The IDB+ module is generated to include information about the data objects and to include an object handler to interact with the data object. Independent claim 1 further recites "registering the component module with an intermediary module." Thus, it is the IDB+ module which is registered with the intermediary module and not the producer component. When a consumer component requires access to the data object, the consumer component sends a request to the intermediary module which forwards the request to the component module, wherein the component module fulfills "the request by providing the requested data object to the consumer component" as recited in independent claim 1.

Burnett fails to disclose a component architecture wherein component modules are used as intermediaries between components (i.e., consumers and producers). Burnett does not teach nor disclose the creation of a component module which contains information identifying a data object, wherein the component module also includes an object handler to interact with a data object or the registering of such a component module with the intermediary module. Specifically, the component module acts as an intermediary between the consumer and the producer objects. The component module receives requests from the consumer component in response to which provides access to the data object for the consumer component.

In contrast, the consumer objects disclosed in Burnett may "call the resource manager's methods *directly* to extract the attributes associated with the object." (*Burnett*, col. 7, lines 65 - 67 (emphasis added)). This illustrates that the consumer objects of Burnett access objects and extract the attributes without the use of an intermediary module as recited in claim 1.

In addition, Burnett does not teach a component module which interacts with the object and provides the object to the consumer component. Burnett only teaches extraction of attributes (e.g., @sys and @host values) from objects and does not disclose a method for consumer components to gain access to the objects themselves. Instead, the consumer components disclosed in Burnett call registered methods to obtain attributes associated with the objects. Burnett specifically states that "the object consumer, which is only interested in the attributes, can obtain them without knowing which resource manager is behind the object or what the object looks like internally." (*Id.*, col. 8, lines 11 - 13). Therefore, the consumer component is not concerned with obtaining the object, only its attributes.

Thus, Burnett neither teaches nor suggests "creating a producer component including a data object and a component module, the component module including information identifying the data object and an object handler to interact with the data object" and "fulfilling the request by providing the requested data object to the consumer component" as recited in claim 1.

Similarly, claim 10 recites "a component module including information identifying a first one of the components and an object handler to interact with a data object, the first one of the components including the data object." Claim 14 recites "each of the producer components including a data object and a component module, the component module including information identifying the data object and an object handler to interact with the data object." Accordingly, applicants respectfully submit that the Examiner should withdraw the rejection under 35 U.S.C. 103(a) of independent claims 1, 10 and 14 and all the claims depending therefrom.

CONCLUSION

In view of the amendments and remarks submitted above, the Applicants respectfully submit that the present case is in condition for allowance. All issues raised by the Examiner have been addressed, and a favorable action on the merits is thus earnestly requested.

Respectfully submitted,

Dated: 1/19/2005 By:__________________

Michael J. Marcin (Reg. No. 48,198)
FAY KAPLUN & MARCIN, LLP
150 Broadway, Suite 702
New York, NY 10038
(212) 619-6000 (phone)
(212) 619-0276 (facsimile)